International Journal of Signal Processing, Image Processing and Pattern Recognition
Vol. 10, No. 6 (2017), pp. 1-12
http//dx.doi.org/10.14257/ijsip.2017.10.6.01

A Novel Algorithm for Color Image Representation Using Non-symmetry and Anti-Packing Model with Squares

Chuanbo Chen[a], Hua Li[a], Zehua Lyu[a]*, Hu Liang[b] and Mudar Sarem[a]

[a]*School of Software Engineering, Huazhong University of Science and Technology,*
Wuhan, 430074, China
[b]*School of Information, Qilu University of Technology, Jinan, 250353, China*
**lvzehua@hust.edu.cn*

Abstract

Color image representation is one of the most important fundamental concepts in image processing. In this paper, a novel algorithm for color image representation using the Non-symmetry and Anti-Packing Model with Squares (NAMS) is proposed. Compared with the existing methods, the proposed algorithm, as far as we know, first manipulates the three channels of color image as a whole in the Non-symmetry and Anti-Packing Model (NAM) theory. Also, Euclidean distance is used to measure the distance weight between two adjacent pixels in CIE-Lab color space instead of comparing three components of the pixels separately. This method can not only preserve the relevance of the color components of the pixels, but also keep the principal component information and reduce information redundancy. What's more, the proposed algorithm can make the sub-patterns fit the contour and the edge self-adaptively because of the non-symmetry. The experimental results show that the NAMS algorithm can get higher reconstruction quality by using fewer sub-patterns than the existing methods, such as the linear quad tree and the original Non-symmetry and Anti-packing Model based method. Therefore, the NAMS algorithm is a better method for color image representation.

Keywords*: color image representation, non-symmetry and anti-packing model, Euclidean distance, CIE-Lab color space*

1. Introduction

In the era of information explosion, the algorithm with high efficiency and low storage has widely requirements in the field of image processing. As a critically important role in image processing issues such as saliency detection [1], visual tracking [2] and object detection [3], an excellent image representation method can be an effective preparation step for the remaining work. Although the color image representation has been investigated widely and many algorithms have been proposed, there still exist some challenges [4-6].

The quad tree representation method is a classical method for image representation. It was first presented by Klinger in the 1970s [7]. Then, Gargantini proposed a linear quad tree representation method and removed the pointers because of the large consumption of the storage [8]. The linear quad tree method is widely used not only in image representation [9-11] but also in many other fields [12-15]. However, it emphasized partitioning the image in a strictly symmetric way which might easily ignore the characteristics of the partitioned blocks. Later, Zheng put forward a color image representation method based on a non-symmetry and anti-packing pattern representation model with the triangle and the square sub-patterns (TSNAM) [16]. He showed that the color image representation method based on the TSNAM can directly reduce the data storage and preserve the characteristics of the blocks more effectively than the linear quad

ISSN: 2005-4254 IJSIP

tree representation method. However, in this method, the color image was decomposed of three gray images (*i.e.*, red, green, and blue) and then represented by using the TSNAM on the three gray images separately. The existing methods split the relevance of the three channels of the color images and lead to errors in the image reconstruction.

In this paper, to improve the representation efficiency of color image, inspired by the optimization idea of the packing problem and the Euclidean distance on the Euclidean norm [17], a novel algorithm for color image representation using the Non-symmetry and Anti-Packing Model with Squares (NAMS) is proposed. The algorithm is based on the CIE-Lab color space, and Euclidean distance is used to measure the perceptual similarity between two adjacent pixels, which can largely preserve the relevance of the color components. The experimental results show that the NAMS algorithm can reduce the data storage much more effectively and the reconstructed images have higher quality than other methods.

The main highlights of this paper are as follows:

Firstly, a novel algorithm for color image representation using NAMS is proposed and it has been proved its representation efficiency theoretically and experimentally.

Secondly, the algorithms of the Non-symmetry and Anti-Packing Model used in color image processing usually regard the color image as three separated gray images, which may ignore the interaction of the three color channels. Considering the relevance, the proposed algorithm first, as far as we know, manipulates the three channels of color image with NAM as a whole.

Thirdly, Euclidean distance is used to measure the distance weight between two adjacent pixels in CIE-Lab color space instead of comparing three components of the pixels separately, which can preserve the relevance of the color components of the pixels. Besides, this method can also keep the principal component information and reduce information redundancy.

Finally, through several measure indicators and multiple sets of contrast experiments, the superiority of the proposed algorithm has been presented clearly not only by statistics but also for the visual perception.

The rest of the paper is organized as follows: In section 2, the idea of the proposed NAMS is introduced. Then, the details of the NAMS algorithm for color image representation are presented in section 3. The experimental results are showed in section 4. And finally, section 5 is the conclusion.

2. Description of NAMS

In this section, the proposed NAMS algorithm for color image representation will be introduced. First, the idea of NAMS will be presented in sub-section 2.1 and then, the NAMS in CIE-Lab space will be proposed and discussed in sub-section 2.2.

2.1. Idea of NAMS

Since the NAMS is based on the NAM which is an anti-packing problem [18-20], the idea of the NAMS can be described as follows: give a color picture (packed pattern) and squares (predefined sub-patterns), pick up these squares (sub-patterns) from the color picture (packed pattern), and then represent the color picture (packed pattern) with the combination of the squares (sub-patterns) [21-23].

By taking two binary images for example, Figure 1 illustrates the idea of the NAMS. Figure 1(a) and (c) are binary images, where the sub-patterns are marked with black pixels. The predefined sub-pattern is square. Since not all of the pixels can be used to construct squares, there are many isolated points left, which are shown in Figure 1(b). By searching every pixel for marking sub-pattern using the raster scanning order, mark out a square with a length of 1, and then enlarge the square to a length of 2. If the larger square is also a standard, it will be marked instead of the smaller square, and the smaller one is turned to be unmarked, as it is illustrated in Figure 1(d).

Since the structure of the anti-packing method is asymmetrical, the non-symmetry model can preserve the characteristics of different blocks, which cannot be achieved by the traditional methods. On the other hand, it can also reduce the amount of the sub-patterns for representing a packed pattern.

2.2. NAMS in CIE-Lab Color Space

Generally, there are two methods to represent a color image. One is simply to turn a color image to a gray image, and then manipulate it as a gray image only. Under this condition, some detailed information of the original color image will be lost. Besides, different ways to reduce the dimensionality will result in different image representations. This may cause many color errors between the reconstructed and the original images in some particular cases.

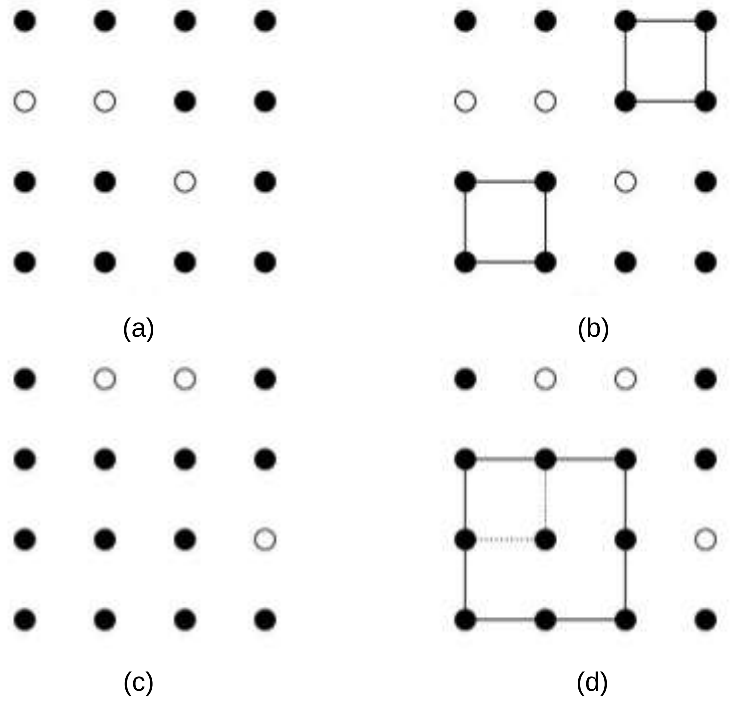


Figure 1. A Simple Illustration for the Idea of NAMS

Another method is to decompose the color image into three gray images, and then handles them separately, and finally composes these three gray images into one color image. Though this method preserves the color information, it will cost three times of the storage area compared with the former one. Besides, manipulating these three gray images separately breaks the relevance of the three components of the pixels, which plays a significant role in presenting the color image. When the three gray images are composed back together, there still exist some errors between the reconstructed image and the original one.

Considering the disadvantages mentioned above, the NAMS algorithm in CIE-Lab color space is proposed. Since the Lab model performs better for human perception and recognition, the Euclidean distance in CIE-Lab color space is chosen to be the key to decide whether the pixels belong to the same sub-pattern or not.

The Euclidean distance between pixel μ and ν used in this paper is computed as follows:

$$w((\mu,\nu)) = \sqrt{(L(\mu)-L(\nu))^2 + (A(\mu)-A(\nu))^2 + (B(\mu)-B(\nu))^2} \quad (1)$$

Without decomposing the color image into three gray images, the method to compute Euclidean distance in CIE-Lab color space preserves the integrity of the characteristics of the color image. On the other hand, the complexity and the storage are largely reduced compared with the method of decomposing color images. As a result, it is a better way to represent a color image for human perception.

3. A Novel Algorithm of NAMS for Color Image Representation

In this section, the encoding and the decoding algorithms will be presented in sub-section 3.1 and 3.2 respectively.

3.1. Encoding Algorithm

For a given color image *I*, an encoding queue set $Q=\{p_1, p_2, \ldots, p_n\}$, where $p_1, p_2, \ldots, p_n$ denote the sub-patterns of the square and n is the total numbers of the instances in *Q*. The following steps present the encoding part of the proposed algorithm:

• Step 1. Change the RGB image *I* to the image *I* in the CIE-Lab color space in order to preserves the relevance of the three channels of the color image.

• Step 2. S*et al*l the pixels in image *I* unmarked.

• Step 3. Set a threshold *K* to decide whether the visited pixels belong to the same sub-pattern or not.

• Step 4. Search the first unmarked pixel of the image *I* in the order of raster scanning, and set the pixel to be the start point (top and left corner in the experiment) of a new square sub-pattern *p*.

• Step 5. Find the biggest sub-pattern *p* to make sure that the spatial weighting of every two pixels in the area of the sub-pattern is less than *K* and mark all the pixels in this sub-pattern.

• Step 6. Record the parameters of the found sub-pattern, *i.e.*, *Q* {*No, s*, *l*}, where *No* is the serial number of the sub-pattern, *s* is the coordinate of the start point of the sub-pattern, and *l* is the length of the square sub-pattern.

• Step 7. Repeat Step 5 and Step 6 until all the pixels are marked in the image *I*. If there exist some isolated pixels which not belong to any sub-patterns, set l = 0, and the start point *s* is the pixel itself.

• Step 8. Out put the encoding queue *Q*.

3.2. Decoding Algorithm

Given an encoding result queue $Q=\{p_1, p_2, \ldots, p_n\}$, a decoding result is stored into a matrix *M* which can represent a color image *I*. The decoding part of the algorithm has the following steps:

• Step 1. Iterate the queue *Q* to achieve the records.

• Step 2. Search the corresponding record to get the coordinate of the start point *s* and the length *l* of the square sub-pattern.

• Step 3. Reconstruct the square sub-pattern using *s* and *l* by giving the average value of all the pixels in the sub-pattern into the matrix *M*.

• Step 4. Repeat Step 2 and Step 3 until all the records in *Q* are searched and reconstructed into the matrix *M*.

• Step 5. Out put the matrix *M* as the pattern of a color image, and then the color image *I* is reconstructed successfully.

4. Complexity and Data Amount Analysis

In this section, the complexity and data amount of the NAMS algorithm are analyzed in subsection 4.1 and 4.2 respectively.

4.1. Complexity Analysis

In this section, for a given color image *I* of *N* pixels, *n* is the number of the sub-patterns of image *I*. For the NAMS algorithm, because of not decomposing the color image into three gray images, the time consumption of coding is in direct proportion to $nN\tau$, where τ is the average segmentation time of the every pixel in the image, and the upper limit is *O* (*log N*). Therefore, the algorithm complexity in the worst circumstance is *O* (*nNlog N*).

As to the space consumption, the space complexity is in direct proportion to the size of the color image, which is *O (N)*.

4.2. Data Amount Analysis

For a given color image *I* of size $2^n \times 2^n \times 3$, to store a triangle, a rectangle, a line segment, and an isolated point takes up *2n + 2, 2n, 2n* and *n* bits, respectively [16]. Suppose that the total sub-pattern numbers of the triangles, rectangles, line segments, and isolated points are M_t , M_r , M_l , and M_p , respectively. For the proposed NAMS algorithm of general sub-patterns (i. e., triangles, rectangles, line segments, and isolated points), the data amount T_{NAMS} can be presented as follows:

$$T_{NAMS} = (2n+2)M_t + 2nM_r + 2nM_l + nM_p \tag{2}$$

For the TSNAM algorithm in [16], the data amount of TSNAM (T_{TSNAM}) and the linear quad tree (LQT) [8] (T_{LQT}) are as follows:

$$T_{TSNAM} = \sum_{i=1}^{3}\sum_{j=0}^{m-1}\left[(2n+2)M_t(i,j) + 2nM_r(i,j) + 2nM_l(i,j) + nM_p(i,j)\right] \tag{3}$$

$$T_{LQT} = \sum_{i=1}^{3}(3n-1+m)N_{LQT}(i) \tag{4}$$

Supposed β is the ratio of the data amount of TSNAM and NAMS, and θ is the ratio of the data amount of LQT and NAMS, β and θ can be presented as follows:

$$\beta = \frac{T_{TSNAM}}{T_{NAMS}} = \frac{\sum_{i=1}^{3}\sum_{j=0}^{m-1}\left[(2n+2)M_t(i,j) + 2nM_r(i,j) + 2nM_l(i,j) + nM_p(i,j)\right]}{(2n+2)M_t + 2nM_r + 2nM_l + nM_p} \tag{5}$$

$$\begin{aligned}\theta = \frac{T_{LQT}}{T_{NAMS}} &= \frac{\sum_{i=1}^{3}(3n-1+m)N_{LQT}(i)}{(2n+2)M_t + 2nM_r + 2nM_l + nM_p} \\ &> \frac{\sum_{i=1}^{3}(3n-1+m)N_{LQT}(i)}{(2n+2)(M_t + M_r + M_l + M_p)}\end{aligned} \tag{6}$$

Since the LQT algorithm and TSNAM algorithm both need to partition the image three times and the NAMS algorithm only need partition once, the number of the sub-patterns ought to be stored is largely decreased. It can come to the conclusion that β >1, θ >1.

As presented above, it is theoretically proved that the proposed NAMS can reduce the number of sub-patterns and the data storage more effectively than the TSNAM algorithm and the LQT algorithm.

5. Experiments

In this section, all of the experiments are performed on a Celeron microprocessor running at 2.6 GHz and 4 GB RAM. The algorithm of NAMS for color image representation has been implemented and compared with TSNAM [16] and LQT [8]. In the experiments, some color images of size 512×512×3 are used.

For better comparison, square is defined as the only sub-pattern type in both the TSNAM and NAMS in the same condition. Since complexities are various for different images, amount of color images have been tested in the experiment and the comparison data of four test color pictures in Figure 2 has been listed in Table 1. Meanwhile, in Figure 3 and 4, two original images (i. e., Lena and Butterfly_1) and their three reconstructed images by using the LQT, the TSNAM and the NAMS methods have been presented respectively. The reconstructed images are filled with the average value of the pixels in each sub-pattern.

Since the property of the non-symmetry of the NAMS method preserves the characteristics of the blocks, the contour of "Lena" shown in Figure 2(d) and the leaves in the background of "Butterfly_1" shown in Figure 3(d) are reconstructed with less artificial artifacts by using the NAMS method than by using the LQT method (as shown in Figure 2(b) and Figure 3(b)). Compared with the reconstructed images by using the TSNAM method (as shown in Figure 2(c) and Figure 3(c)), the images reconstructed by using the NAMS (as shown in Figure 2(d) and Figure 3(d)) have less color errors compared to the original images. This is due to the benefit of keeping the relevance of the color components without manipulating them separately.

(a) Lena

(b) Flower

(c) Butterfly_1 (d) Butterfly_2

Figure 2. The Four Test Original Color Image

Table 1 has shown the comparison of the number of the sub-patterns (the blocks), the Mean Square Error (MSE), the Peak Signal to Noise Ratio (PSNR), and structural similarity index measurement (SSIM) of the LQT, TSNAM, and NAMS methods, which can be defined as follows:

$$PSNR = 10\log_{10}\frac{255^2}{MSE} \tag{7}$$

where MSE is the mean square error between the original image X and the reconstructed image, which can be defined as follows:

$$MSE = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}\left(X_{ij} - \bar{X}_{ij}\right)^2}{MN} \tag{8}$$

(a) Original Image (b) Reconstructed Image by LQT

(c) Reconstructed Image by TSNAM

(d) Reconstructed Image by NAMS

Figure 3. The Comparison of Reconstructed Images of Lena

SSIM can be defined as follows:

$$SSIM = \frac{\left(2\mu_x\mu_y + C_1\right)\left(2\sigma_{xy} + C_2\right)}{\left(\mu_x^2 + \mu_y^2 + C_1\right)\left(\sigma_x^2 + \sigma_y^2 + C_2\right)} \tag{9}$$

where (x, y) is the coordinate of one pixel, μ is the average value of two pixels in the two compared images, σ is the standard deviation. C_1, C_2 are predefined parameters. In the experiments, C_1 is defined as 0.01 and C_2 is 0.03.

(a) Original Image

(b) Reconstructed Image by LQT

(c) Reconstructed Image by TSNAM (d) Reconstructed Image by NAMS

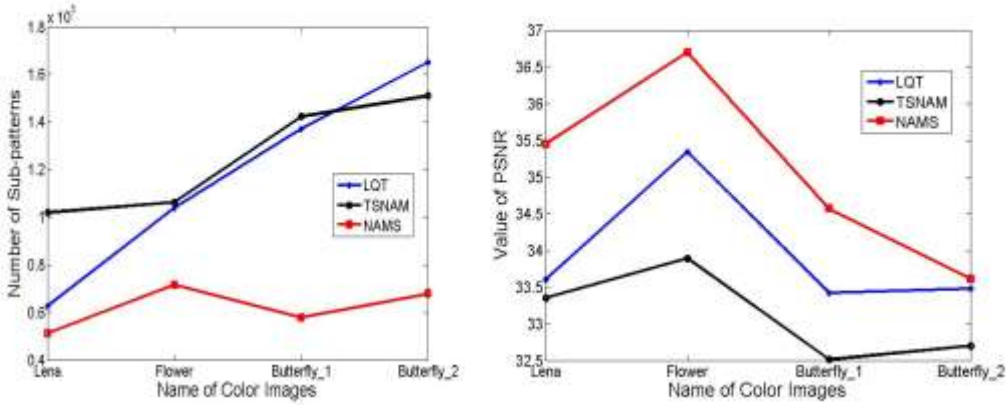
Figure 4. The Comparison of Reconstructed Images of Butterfly_1

Besides, Figure 5 has shown the line graphs of the comparison of the number of the sub-patterns (include isolated pixels) of the NAMS algorithm, the LQT algorithm, and the TSNAM algorithm.

According to Table 1 and Figure 5, the proposed NAMS algorithm has obviously fewer blocks than other methods (i. e., the LQT and the TSNAM) while the PSNR value in the proposed NAMS algorithm is higher and the MSE value is lower. Besides, the SSIM values of the NAMS algorithm in the four images are all above 0.9 and higher than others. As a result, the NAMS algorithm can represent the color images more efficiently with less storage space than the LQT and the TSNAM algorithms. And thus, it is a better way to represent a color image.

Table 1. The Comparison of Experimental Data between LDT, TSNAM, and NAMS

	Lena				Flower			
	blocks	MSE	PSNR	SSIM	blocks	MSE	PSNR	SSIM
LQT	62679	28.39	33.60	0.8962	103908	19.00	35.34	0.9065
TSNAM	102018	30.04	33.35	0.8725	106360	26.55	33.89	0.8545
NAMS	51293	18.52	35.45	**0.9375**	71852	13.92	36.70	**0.9362**
NAMS/LQT	**0.8183**	**0.6524**	**1.0552**	**1.0461**	**0.6915**	**0.7326**	**1.0382**	**1.0328**
NAMS/TSNAM	**0.5028**	**0.6166**	**1.0630**	**1.0745**	**0.6756**	**0.5242**	**1.0828**	**1.0956**
	Butterfly_1				Butterfly_2			
	blocks	MSE	PSNR	SSIM	blocks	MSE	PSNR	SSIM
LQT	137073	29.58	33.42	0.8792	165183	29.16	33.48	0.9113
TSNAM	142338	36.47	32.51	0.8741	151113	34.92	32.70	0.8885
NAMS	57893	22.71	34.57	**0.9256**	67829	28.30	33.61	**0.9171**
NAMS/LQT	**0.4224**	**0.7678**	**1.0343**	**1.0527**	**0.4106**	**0.9705**	**1.0039**	**1.0064**
NAMS/TSNAM	**0.4067**	**0.6226**	**1.0633**	**1.0589**	**0.4489**	**0.8106**	**1.0279**	**1.0322**



(a) Number of Sub-patterns (b) Value of PSNR

Figure 5. Line Graphs of the Comparisons of LQT, TSNAM, and NAMS

6. Conclusion

In this paper, inspired by the relevance of the three components of each pixel of a color image, a novel algorithm for color image representation by using the NAMS algorithm has been proposed. By comparing the representation algorithm of NAMS with those of TSNAM and LQT, it has been proved that the proposed NAMS algorithm for color images can save the storage space much more effectively as well as reconstruct color images better than the other methods in both the PSNR values and the visual quality. Therefore, it is a better method to represent color images. This present work gives an improvement for color image representation by preserving the relevance of the three components of the pixels in color images.

References

[1] Z. Liang, Z. Chi, H. Fu, D. Feng. Salient object detection using content-sensitive hypergraph representation and partitioning. Pattern Recognition, vol. 45, no. 11, **(2012)**, pp. 3886-901.

[2] X. Lan, A. J. Ma, P. C. Yuen and R. Chellappa, "Joint Sparse Representation and Robust Feature-Level Fusion for Multi-Cue Visual Tracking", IEEE Transactions on Image Processing, vol. 24, no. 12, **(2015)**, pp. 5826-41.

[3] X. Zhou, C. Yang and W. Yu, "Moving Object Detection by Detecting Contiguous Outliers in the Low-Rank Representation", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 3, **(2013)**, pp. 597-610.

[4] K. S. Gurumoorthy, A. Rajwade, A. Banerjee and A. Rangarajan, "A method for compact image representation using sparse matrix and tensor projections onto exemplar orthonormal bases", IEEE Transactions on Image Processing, vol. 19, no. 2, **(2010)**, pp.322–334.

[5] Y. Li, S. Wang, Q. Tian and X. Ding, "Feature representation for statistical-learning-based object detection: A review", Pattern Recognition, vol. 48, no. 11, **(2015)**pp. 3542-3559.

[6] M. D. Adams, "A highly-effective incremental/decremental Delaunay mesh-generation strategy for image representation", Signal Processing, vol. 93, no. 4, **(2013)**, pp.749-64.

[7] A. Klinger, "Data structure and pattern recognition", In: Proceedings of International Joint Conference on Pattern Recognition (IJCPR), Washington, D.C., USA, **(1973)**, pp.497-498.

[8] I. Gargantini, "An effective way to represent quadtrees", Communications of the ACM, vol. 25, no. 12, **(1982)**, pp. 905–910.

[9] J. M. Rodrigues and C. G. Soares, "Exact pressure integrations on submerged bodies in waves using a quadtree adaptive mesh algorithm", International Journal for Numerical Methods in Fluids, vol. 76, no. 10, **(2014)**, pp.632-52.

[10] R. Ngadiran, S. Boussakta, A. Bouridane and F. Khelifi, "Exploiting chrominance planes similarity on listless quadtree coders", IET Image Processing, vol. 7, no. 6, **(2013)**, pp.575-85.

[11] C. Zhang, Y. Zhang, W. Zhang and X. Lin, "Inverted linear quadtree: Efficient top k spatial keyword search", 2013 29th IEEE International Conference on Data Engineering (ICDE 2013), in Brisbane, Australia, **(2013)**, pp.901-12.

[12] Y. Han and Y. Byun, "Automatic and accurate registration of VHR optical and SAR images using a quadtree structure", International Journal of Remote Sensing, vol. 36, no. 9, **(2015)**, pp.2277-2295.
[13] I. Šimeek, D. Langr and J. Trdlika, "Efficient converting of large sparse matrices to quadtree format", Proceedings - 16th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, SYNASC 2014, in Romania, **(2015)**, pp. 122-129.
[14] Y. Liu, O. Deforges, F. Pasteau and K. Samrouth, "One pass quality control and low complexity RDO in a quadtree based scalable image coder", 2013 IEEE Second International Conference on Image Information Processing (ICIIP-2013), in India, **(2013)**, pp. 187-92.
[15] M. Ding and R. Tong, "Efficient dark channel based image dehazing using quadtrees", Science China Information Sciences, vol. 56, no. 9, **(2013)**, pp. 092120 -129.
[16] Y. Zheng and J. Zhang, "A Gray-code-based color image representation method using TSNAM", CAAI Transactions on Intelligent Systems, vol. 7, no. 2, **(2012)**, pp.142–147.
[17] D. A. Klein and S. Frintrop, "Salient pattern detection using W2 on multivariate normal distributions", In Proc. Joint 34th DAGM-OAGM Symp. **(2012)**, pp.246–255.
[18] C. Chen and D. He, "Heuristic method for solving triangle packing problem", Journal of Zhejiang University, vol. 6, no. 6, **(2005)**, pp. 565–570.
[19] C. Chen, Y. Zheng and M. Sarem, "A novel non-symmetry and anti-packing model for image representation", Chinese Journal of Electronics, vol. 18, no. 1, **(2009)**, pp.89–94.
[20] Y. Zheng and M. Sarem, "A fast algorithm for computing moments of gray images based on NAM and extended shading approach", Frontiers of Computer Science in China, vol. 5, no. 1, **(2011)**, pp. 57–65.
[21] Y. Zheng, W. Zhou, X. Xie and X. Mo, "A fast algorithm for computing the lower order moments of binary images by using non-symmetry and anti-packing model", Journal of Computational Information Systems, vol. 9, no. 5, **(2013)**, pp.1949-1956.
[22] Y. Zheng, W. Zhou and X. Mo, "A new NAM-based algorithm for computing Hu moments of binary images", Journal of Information and Computational Science, vol. 10, no. 8, **(2013)**, pp. 2481-2488.
[23] Y. Zheng and M. Sarem, "A novel binary image representation algorithm by using NAM and coordinate encoding procedure and its application to area calculation", Frontiers of Computer Science, vol. 8, no. 5, **(2014)**, pp.763-772.

Authors

Chuanbo Chen, he received Ph.D. degree from Huazhong University of Science and Technology, Wuhan, China. He is currently a professor in School of Software Engineering, Huazhong University of Science and Technology, Wuhan, China. His currently research interests include Image Processing and AI.

Hua Li, she is a PhD Candidate in the School of Software Engineering, Huazhong University of Science and Technology, Wuhan, China. Her reach interests include Pattern Recognition and Image Processing.

Zehua Lyu, he works as a Lecturer in the School of Software Engineering in Huazhong University of Science and Technology. He received his Ph.D. Degree in Huazhong University of Science and Technology in 2008. His research interests include Image Processing and AI.